

Event Management App Using Angular and JSON

Home.html

```
<!DOCTYPE html>
<html lang="en">
<body>
  <div class="container mt-5"><textarea class="form-control"
rows="12" placeholder="Home works!"></textarea></div>
</body>
</html>
```

Appcomponent.html

```
<head>
  <style>
    .ml-auto {
      margin-left: auto !important;
    }
  </style>
</head>
<div class="container bg-">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <h2><a class="navbar-brand bg-info rounded" style="font-size:
larger; padding: 10px;" href="/">Portal</a></h2>
    <a class="navbar-brand bg-warning rounded" style="padding: 8px;"
href="/home">Home</a>
    <a class="navbar-brand bg-primary rounded" style="padding: 8px;"
href="/employees">Employees</a>
    <form class="form-inline ml-auto" action="#">
      <input class="form-control mr-sm-2" type="search"
placeholder="Search" aria-label="Search">
      <a href="/employeesdetails" class="btn btn-outline-success
my-2 my-sm-0" type="submit">Search</a>
    </form>
  </nav>
</div>
<router-outlet></router-outlet>
```

Employees.html

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <style>
    .ml-auto {
      margin-left: auto !important;
    }
  </style>
</head>
<body>
  <div class="container">
    <!-- the employess details -->
    <div class="container mt-5">
      <div class="row">
        <div class="col-6">
          <a href="/addemp" class="btn btn-primary">Add Employee</a>
        </div>
      </div>
      <div *ngFor="let employee of empl">
        <p>ID: {{ employee.id }}</p>
        <p>First Name: {{ employee.first_name }}</p>
        <p>Last Name: {{ employee.last_name }}</p>
        <p>Email: {{ employee.email }}</p>
        <a href="/employeesdetails" class="btn btn-info" >View
Details</a><hr>
      </div>
    </div>
  </div>
</body>
</html>

```

Employeeclass

```

export class Employee {
  id:number;
  first_name:string;
  last_name:string;
  email:string;

  constructor(id:number,first_name:string,last_name:string,email:string)
  {
    this.id=id;
    this.first_name=first_name;this.last_name=last_name;this.email=email;
  }
}

```

EmployeeService

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { Employee } from '../classes/employee';

@Injectable({
  providedIn: 'root'
})
export class EmpService {

  private jurl="http://localhost:3000";
  constructor(private http:HttpClient) { }

  //getting all employees
  getAllDetails():Observable<Employee[]> {
    return this.http.get<Employee[]>(`${this.jurl}/Employee`);
  }

  //adding a employee

  addEmployee(id:number,first_name:string,last_name:string,email:string):Observable<Employee> {
    const newempl:Employee={id: id,first_name: first_name,last_name:
last_name,email: email}
    return this.http.post<Employee>(`${this.jurl}/Employee`, newempl);
  }

  //Deleting an Employee details
  delEmployee(id:number):Observable<Employee> {
    return this.http.delete<Employee>(`${this.jurl}/Employee/${id}`);
  }

  //Searching
  searchEmployee(id:number):Observable<Employee> {
    return this.http.get<Employee>(`${this.jurl}/Employee/${id}`);
  }
}
```

Employee.ts

```
export class EmployeesComponent implements OnInit {

  empl!:Employee[];
```

```

    constructor(private empService:EmpService) {}

ngOnInit() {
    //getting all employee details
    this.empService.getAllDetails().subscribe((response)=>{this.empl=response;}, (error)=>{console.log(error)});
}

```

Add-employee.ts

```

export class AddEmployeeComponent {

    empl:Employee ={
        id: 0,
        first_name: '',
        last_name: '',
        email: ''
    }

    constructor(private empService:EmpService) { }

    onSubmit() {

        this.empService.addEmployee(this.empl.id,this.empl.first_name,this.empl.last_name,this.empl.email).subscribe((response)=>{this.empl=response;}, (error)=>{console.log(error)});
        alert(`${this.empl.first_name} Has been added!`);
    }
}

```

Del-employee.ts

```

export class DelEmpComponent {

    empl!:Employee;
    idnumber!: number;
    constructor(private empService:EmpService) { }

    onSubmit() {

        this.empService.delEmployee(Number(this.idnumber)).subscribe((response)=>{this.empl=response;}, (error)=>{console.log(error)});
        alert("Deleted");
    }
}

```

```
}
```

Employee-details.ts

```
export class EmployeeDetailsComponent implements OnInit {

    empl!:Employee;
    id!:number;

    constructor(private empService:EmpService) { }

    ngOnInit(): void {

this.empService.searchEmployee(this.id).subscribe((response)=>{this.empli=response;}, (error)=>{console.log(error);});

    }
}
```

App.modules.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HomeComponent } from './home/home.component';
import { EmployeesComponent } from './employees/employees.component';
import { EmployeeDetailsComponent } from './employee-details/employee-details.component';
import { EmpService } from './service/emp.service';
import { HttpClientModule } from '@angular/common/http';
import { AddEmployeeComponent } from './add-employee/add-employee.component';
import { FormsModule } from '@angular/forms';
import { DelEmpComponent } from './del-emp/del-emp.component';

@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    EmployeesComponent,
    EmployeeDetailsComponent,
    AddEmployeeComponent,
    DelEmpComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [EmpService],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

```

    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        HttpClientModule,
        FormsModule
    ],
    providers: [EmpService],
    bootstrap: [AppComponent]
  })
}
export class AppModule { }

```

Routing-paths

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AppComponent } from './app.component';
import { EmployeesComponent } from './employees/employees.component';
import { EmployeeDetailsComponent } from
'./employee-details/employee-details.component';
import { AddEmployeeComponent } from
'./add-employee/add-employee.component';
import { DelEmpComponent } from './del-emp/del-emp.component';
import { HomeComponent } from './home/home.component';

const routes: Routes = [
    {path:"",component:AppComponent},
    {path:"home",component:HomeComponent},
    {path:"employees",component:EmployeesComponent},
    {path:"employeesdetails",component:EmployeeDetailsComponent},
    {path:"addemp",component:AddEmployeeComponent},
    {path:"delemp",component:DelEmpComponent}
];

@NgModule({
    imports: [RouterModule.forRoot(routes)],
    exports: [RouterModule]
})
export class AppRoutingModule { }

```

✓ PHASE-4

✓ Event-Managment-app

> .angular

> .vscode

> node_modules

✓ server

{ } db.json

TS server.ts

✓ src

✓ app

> add-employee

> classes

> del-emp

> employee-details

> employees

> home

> service

TS app-routing.mod...

app.component.css

<> app.component.h...

TS app.component.s...

TS app.component.ts

TS app.module.ts

> assets

★ favicon.ico

<> index.html

TS main.ts

styles.css

⚙ .editorconfig

≡ .gitignore

{ } angular.json

← ↻ ⓘ localhost:4200/home

A ☆ 📄 ☆ 📄 📄 🌐 ... b

Portal

Home

Employees

Search

Search

Home works!

← ↻ ⓘ localhost:4200/employees

A ☆ 📄 ☆ 📄 📄 🌐 ... b

Portal

Home

Employees

Search

Search

Add Employee

ID: 1

First Name: Sebastian

Last Name: Eschweiler

Email: sebastian@codingthesmartway.com

View Details

ID: 2

First Name: Steve

Last Name: Palmer

Email: steve@codingthesmartway.com

View Details

ID: 3

First Name: Ann

Last Name: Smith

1

Search

Details of Sebastian

• Employee ID: 1
• Employee FirstName: Sebastian
• Employee LastName: Eschweiler
• Employee Email: sebastian@codingthesmartway.com

Employee List

Delete

Update

Portal

Home

Employees

localhost:4200 says
Suzie Has been added!

OK

Search

Search

Emp-ID

5

First-Name

Suzie

Last-Name

Zane

Email

suzane@email.com

ADD

Back

←

↺

localhost:4200/employees

View Details

ID: 3

First Name: Ann

Last Name: Smith

Email: ann@codingthesmartway.com

View Details

ID: 4

First Name: Jhon

Last Name: Doe

Email: jhondoe@email.com

View Details

ID: 5

First Name: Suzie

Last Name: Zane

Email: suzane@email.com

View Details

←

↺

localhost:4200/delemp

Aa

☆

🔍

🔖

🔗

Portal

Home

Employees

Search

Search

Enter the ID

Emp-ID

5

Delete

Back

localhost:4200 says

Deleted

OK

After Deletion in Postman

The image shows the Postman interface for a GET request to `http://localhost:3000/Employee`. The request is successful, returning a 200 OK status with a response time of 109 ms and a body size of 868 B. The response body is displayed in the 'Body' tab, showing a JSON array of four employee objects. The 'Pretty' view is selected, and the 'JSON' format is chosen.

API: api using angular / Event-json

Method: GET URL: http://localhost:3000/Employee

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (13) Test Results

200 OK 109 ms 868 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "first_name": "Sebastian",
5     "last_name": "Eschweiler",
6     "email": "sebastian@codingthesmartway.com"
7   },
8   {
9     "id": 2,
10    "first_name": "Steve",
11    "last_name": "Palmer",
12    "email": "steve@codingthesmartway.com"
13  },
14  {
15    "id": 3,
16    "first_name": "Ann",
17    "last_name": "Smith",
18    "email": "ann@codingthesmartway.com"
19  },
20  {
21    "id": 4,
22    "first_name": "Jhon",
23    "last_name": "Doe",
24    "email": "jhondoe@email.com"
25  }
26 }
```

[View Details](#)

ID: 2

First Name: Steve

Last Name: Palmer

Email: steve@codingthesmartway.com

[View Details](#)

ID: 3

First Name: Ann

Last Name: Smith

Email: ann@codingthesmartway.com

[View Details](#)

ID: 4

First Name: Jhon

Last Name: Doe

Email: jhondoe@email.com

[View Details](#)
